RL-TM-92-8
In-House Report
February 1992

THE EVOLVING DATA DICTIONARY

Patrick K. McCabe

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TM-92-8 has been reviewed and is approved for publication.

APPROVED: John R. Bratge

JOHN R. BRATGE, Chief
Intellignece Data Handling Division

APPROVED: Thaddeus Domurat

THADEUS J. DOMURAT
Technical Director
Intelligence & Reconnaissance Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL (IRDD), Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1992	

4. TITLE AND SUBTITLE
THE EVOLVING DATA DICTIONARY

5. FUNDING NUMBERS
PR - 4594
TA - PR
WU - OJ

6. AUTHOR(S)
Patrick K. McCabe

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (IRDD)
Griffiss AFB NY 13441-5700

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TM-92-8

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (IRDD)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Patrick K. McCabe/IRDD/(315) 330-2171

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

System Analysts have used a variety of techniques to define, implement, and document information systems. As databases grew large, it became increasingly difficult, if not impossible, to determine what data elements were used by what applications. In many cases, as needs changed, it was easier to build a new application with unique data elements than it was to try and find existing data elements that could be used. Data dictionaries were built to solve this problem. Originally hardcopy references, data dictionaries are evolving into fully automated, in-line active control mechanisms. Data dictionary functionality continues to evolve in the direction of heterogeneous sy system access support and information management.

14. SUBJECT TERMS
Data Dictionary, Database, Systems Analysis
Information Resource Dictionary System

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	U/L

The Origin of the Data Dictionary

System Analysts have used a variety of techniques to define, implement, and document information systems. My personal favorite was one called the "Brown Freezer Paper" (BFP) technique that I heard about in a systems analysis course. BFP consisted of getting a big brown roll of freezer paper, pasting a copy of every form used by an organization to the paper, and drawing lines (either with crayons or magic markers) between all blocks on all forms that required the same information. The employee name block on each form would be connected by a common line for example.

BFP was primarily used to implement database management systems in organizations that had no automated support. Today, most organizations have had automated support for some time. Database management systems eliminated the need for techniques as unwieldy as BFP, but gave rise to other problems. Data elements were managed by the database management system, but the definitions were "hardwired" in the applications themselves. As databases grew large, it became increasingly difficult, if not impossible, to determine what data elements were used by what applications. In many cases, as needs changed, it was easier to build a new application with unique data elements than it was to try and find existing data elements that could be used.

It became apparent at this stage of the game that the definitions of corporate data, as well as data, were critical corporate resources. Data dictionaries were built to solve this problem. Initially, the data dictionary was a hardcopy reference. Manual data dictionaries suffered from two drawbacks however, they were unwieldy to use and difficult to update. It was easy to not get around to updating the data dictionary while the Information System staff was in continuous combat mode trying to satisfy user application demands.

Without a mechanism to force updates to the dictionary in the course of development and maintenance, the dictionary quickly became so out of step with the implementation as to become useless. Procedural methodologies to prevent this were time consuming and expensive in terms of manpower hours and schedule requirements. The ideal is to provide an environment for application and database developers and implementors that force them to go through the data dictionary during development. This approach would ensure that existing data element definitions would be used to satisfy new requirements without necessitating new application development. Additionally, the currency of

the dictionary would be guaranteed. The elimination of duplicative application development gives organizations greater control over their information resources, and substantial savings in operations and maintenance costs.

But what constitutes a data element definition? Initially, data element definitions consisted of formats; i.e. how many alphanumeric characters, leading dollar signs, fixed number of decimal places, and so on. Formats only went so far however. Which applications used which data elements? Who was responsible for the creation of the data element and why? What about applications and documentation? Couldn't data elements representing applications (source and object modules), data, and documentation be created, controlled and maintained by the dictionary? What life-cycle phase is a particular application or data element in? Current state of the art in data dictionaries place the data dictionary in the middle of the data element definition process.

The data dictionary has changed from a passive reference mechanism, to the foundation of a comprehensive, active development environment; the glue that controls applications, data, documentation, and the interrelationships among each. In a very real sense, the data dictionary function has become more important than the database management function. However, most data dictionary implementations are tightly coupled to specific DBMSs or to DBMS's in general.

The Information Resource Dictionary System

As discussed above, benefits of an active, in-line data dictionary include improved identification of existing information resources, reduction of application development, simplified software and data conversion, consistent documentation, and increased portability of acquired skills with corresponding decreases in training costs. What was lacking was an agreed upon definition of what services would be provided by the data dictionary. Additionally, a standardized interface making provision of data dictionary services vendor independent, was not available. The National Institute of Standards (NIST) initiated an effort to standardize the interface to data dictionaries. The result was ANSI Standard X3.138-1988, the Information Resource Dictionary System (IRDS).

A preliminary cost benefit analysis done in preparation for work on the IRDS, estimated that savings to the Federal Government could reach

$120 Million in constant 1983 dollars by the early 1990's.[1] Additionally, the IRDS can be used to aid development, modification, and maintenance of manual and automate systems throughout their life cycle, support an organization-defined data element standardization program, support records, reports and forms management, spanning the range from manual to fully automated environments.

The IRDS is implemented utilizing the entity-relationship model. Information is described in terms of entities (specific objects, real or abstract, about which data are stored), relationships (an association among entities) and attributes (a named characteristic or descriptor of an entity or relationship). Architecturally, the IRDS consists of four layers (see figure 1); information resources, the information resource dictionary (IRD), the information resource dictionary schema, and the information resource dictionary schema description. The IRDS standard specifies the syntax and semantics of a command language that operates against the IRD and IRD Schema. Command language syntax is specified in a Backus Naur Form(BNF) like form, while semantics are presented as a set of actions and rules. [2]

The information resources layer consists of the contents of a corporate resource base. The corporate resource base consists of the contents of all the databases that are used and controlled by the corporation that owns the databases. The standard does not address the content of corporate databases, it provides a mechanism to describe that data.[3]

The Information Resource Dictionary (IRD) layer describes the objects and associations among objects at the information resource level. Object descriptions are referred to as entities and association descriptions are referred to as relationships. Attributes are properties of entities and relationships. It is an important distinction that relationships are directed associations between entities. Another extremely important concept to

---

1 Alan Goldfine and Patricia Konig, A Technical Overview of the Information Resource Dictionary System, Center for Programming Science and Technology, National Institute of Standards and Technology, April 1985, pp 4.

2 American National Standard for Information Systems, Information Resource Dictionary System, ANSI X3-138-1988, National Institute of Standards and Technology, July 1988, p.7

3 Ibid, p. 4

the standard at this layer of the architecture is the concept of "type". "Type" defines a range of values an object may take (e.g. an object of type "integer" may take any non-fractional value). Each entity and relationship are of unique type, with associated attribute types and attribute group types (which are ordered sets of attribute types). [4]

The Information Resource Dictionary Schema (IRD Schema) layer describes the content and structure of the IRD. A description of the entity type, relationship type, attribute type, and attribute group type for every entity, relationship, attribute and attribute group in the IRD is contained in this layer. Figure 2 illustrates the associations among meta-entities (rectangles) and meta-relationships (hexagons). The upper half of the hexagon names the forward meta-relationship type and the lower half names the inverse meta-relationship. The portion of the schema surrounded by the dotted line is defined in module 4 of the standard and provides the basis for life cycle management of the contents of the IRD. Another point to note from figure 2 is that a number of meta-entity types do not participate in meta-relationships; IRD Partition, Quality-Indicator, IRDS-Limits, IRDS Defaults, IRDS Reserved Names, and Names. These meta-entities represent conditions of entities or processing constraints associated with the IRDS and its interpretation of the IRD Schema content.[5]

The IRD Schema Definition layer contains the types of all objects which can be defined in IRD Schema, the types of all relationships that can exist among those objects, and certain properties of both. The function of this layer is to support the extension or modification of an installation's IRD Schema. The standard does not specify the content of the IRD Schema Definition layer in order to avoid "unnecessary complexity".[6]

Therefore, the Information Resource Dictionary System (IRDS) Standard specifies two layers of description: the IRD, which contains the metadata describing information resources; and the Dictionary Schema, which models the IRD. Integrity rules can be stored in the IRD and in the IRD Schema to control the utilization of the metadata describing information resources and the definitions of those resources. The Dictionary Schema can be extended to reflect changes in the system environment, facilitating introduction of new database management

---

4 Ibid, p. 7

5 Ibid, p. 4

6 Ibid, p. 3,6

systems. Facilities to partition the dictionary and to control the content and entity relationships within each partition are provided by the

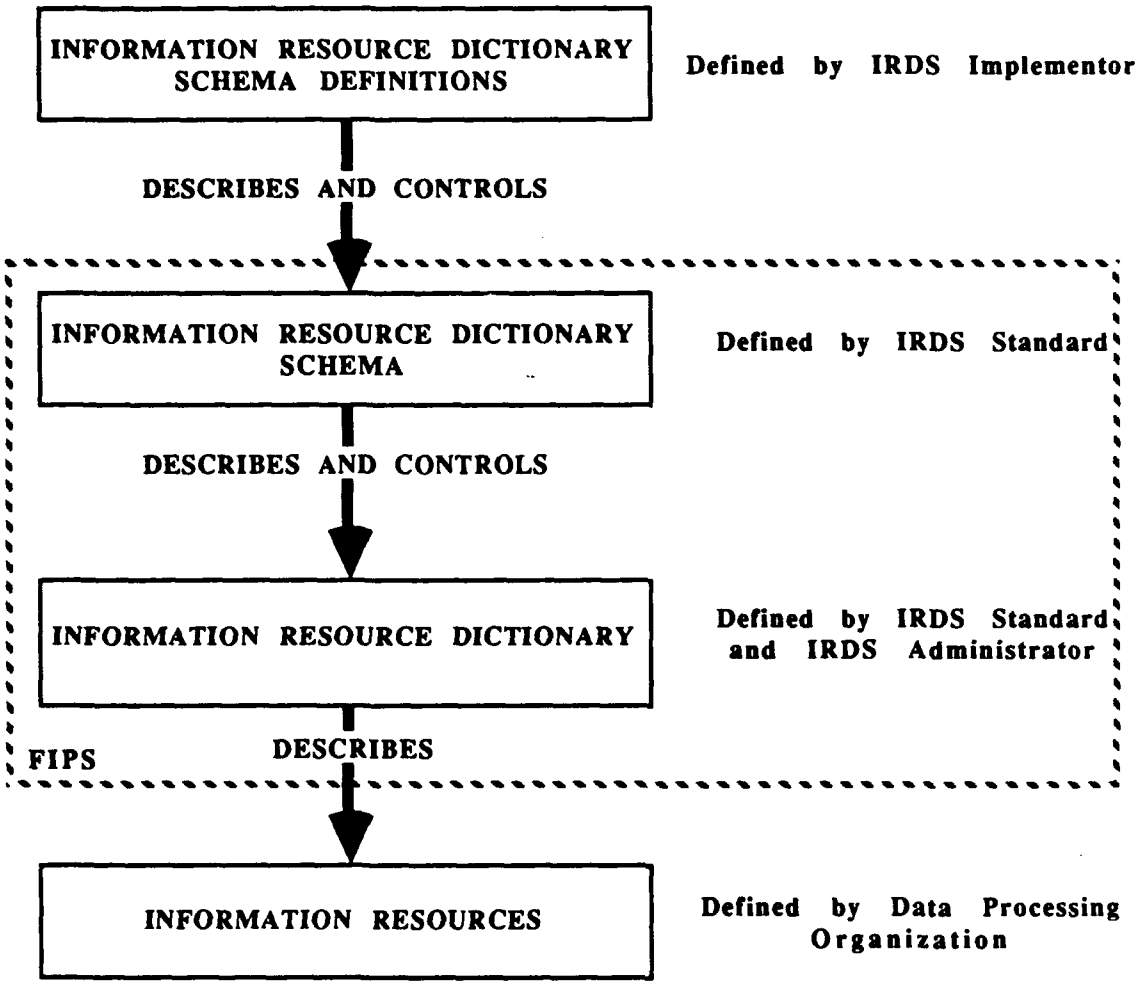

Figure 1

Information Resource Dictionary Architecture

standard. Also, the facilities support the implementation of versions of the database dedicated to production, maintenance, test and development. [7]

---

[7] AOG Systems Corporation, Design Concepts for Database Utilities, RADC TR-86-48, pp. 15-17.

Data Dictionary Extensions

In parallel with, yet unrelated to the development of the IRDS standard, Rome Laboratory began investigating the heterogeneous database access problem. RL's motivation was the excruciating difficulty experienced during the conversion of an operational information system from a home-grown database management system on a Honeywell hardware configuration to a commercial-off-the-shelf (COTS) database management system on an IBM hardware configuration. During the course of the transition it was essential that system users be provided access to the data, regardless of which system it was on, and at the same time, be insulated from the mechanics of database navigation.

Intelligence analysts continue to require access to an ever increasing number of diverse information systems to effectively support their mission. Data communications standards have made connectivity to these systems a manageable engineering problem. However, data communications standards and the recent emphasis on open system architectures does not go far enough.

Each distinct intelligence information system is built around a database designed to support a functionally unique set of analytical functions. Even though the input to each system might be the same, the way in which the data are used and processed, dictates different database management system requirements, characteristics and unique database structures. Consequently, the analyst must retain a comprehensive mental picture of, data availability, location, and database organization, as well as the mechanics of database navigation, for several database systems.

This problem has been addressed, if at all, by special purpose application software containing information describing what data is available from where, and how to get it. This approach requires a very small, well defined, and carefully partitioned set of databases. Minor changes to data elements or to applications result in unpredictable side effects to operational software. This problem is severely exacerbated in a network environment when multiple database systems are involved.

A more robust approach is to place definitions of information resources in a logically centralized repository at the network level. This approach immediately simplifies application software, eliminating application code that is sensitive to change, and maintaining traceability among a wide range of application and information objects. The IRDS standard, undergoing finalization at the time, was ideal as the required

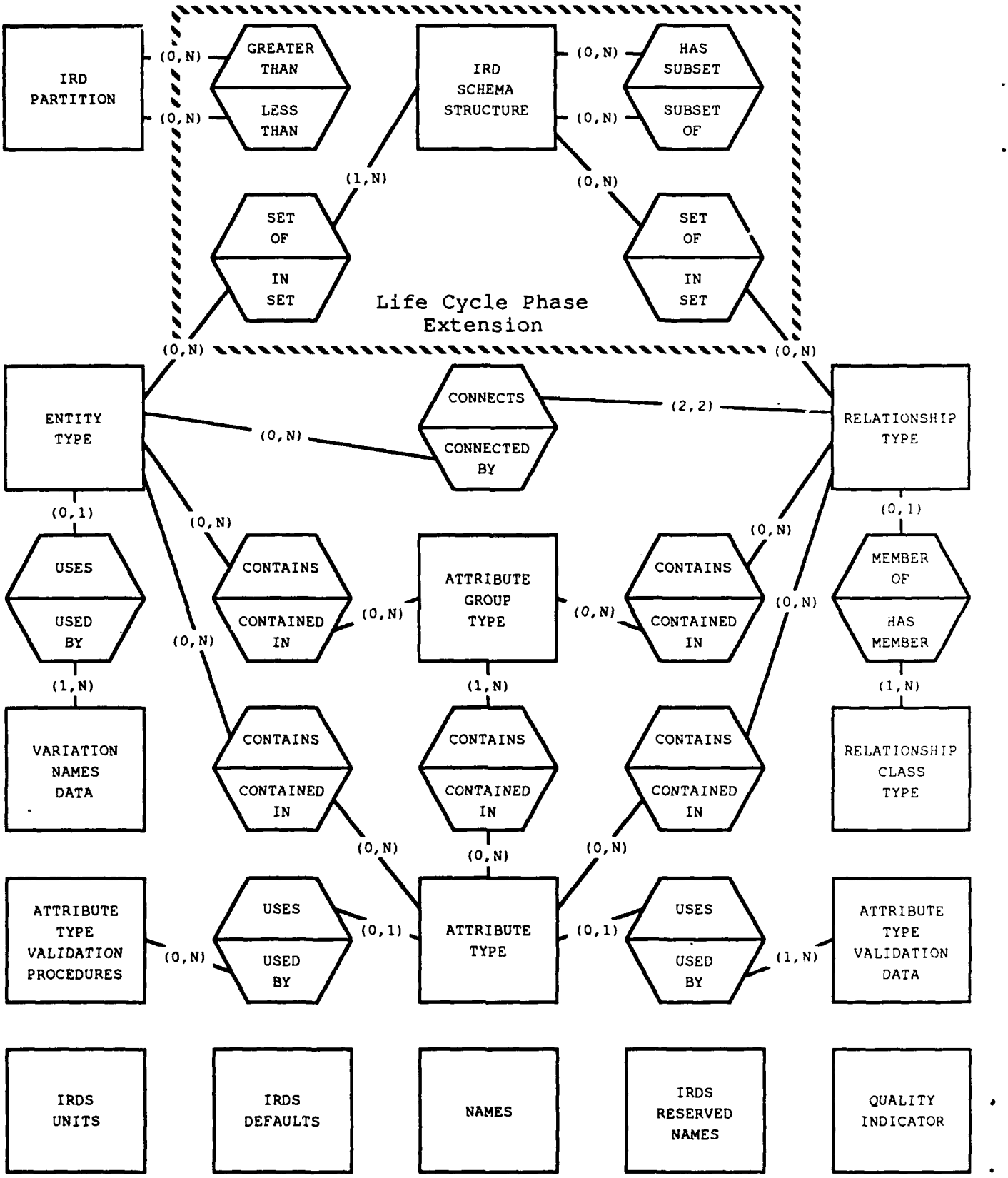

Figure 2

Associations of IRD Schema Entities

network level repository of information resource definitions. By making the implementation of the IRDS an active, in-line capability, access to the database(s) would take place via the services of the IRDS. Whether for data retrieval, application development, database administration, configuration management, or quantitative impact assessment of proposed changes, the services provided by the IRDS play a vital role in controlling the information resource management process as well as information resources. The result is a robust mechanism that supports effective management of information system evolution.

The IRDS is essentially a database of "metadata". Providing this function at the network level allows system and application independent access to data. Real time validation and integrity checking of incoming data can be performed both as foreground and background processes. An additional benefit is the on-line maintenance of interrelationships among data elements, data structures, file structures, applications, and on-line documentation. This capability provides users and developers with a comprehensive picture of what data elements are available and under what constraints. Accurate, quantifiable impact (for multiple systems at the network level) assessments can be made when data element, application, or requirement modifications are proposed. Proliferation of applications and data elements can be controlled. Integration of new capabilities or systems would be easier.

The information resource dictionary system provides a solid foundation on which to provide in-line data dictionary services at the network level. Given the nature of these services, it makes sense to provide them as a network resource in a dedicated server referred to as the Database Query Support Processor (QSP). Other mechanisms in the server will utilize meta descriptions of the constituent databases to interact with the user to formulate queries and generate response strategies to those queries against multiple databases.

There are four functional relationships between the Database Query Support Processor (QSP), the user, and the databases in a network environment (see figure 3). Functional relationship 1 is the interaction between a user developing a generalized, ad-hoc query, or an application issuing a formatted query and the services provided by the QSP. The result of this interaction is the formulation of a syntactically correct, semantically meaningful query that is decomposed into subqueries. A query/subquery execution strategy is developed, a response composition strategy is developed, and the metadata required to track and manage the

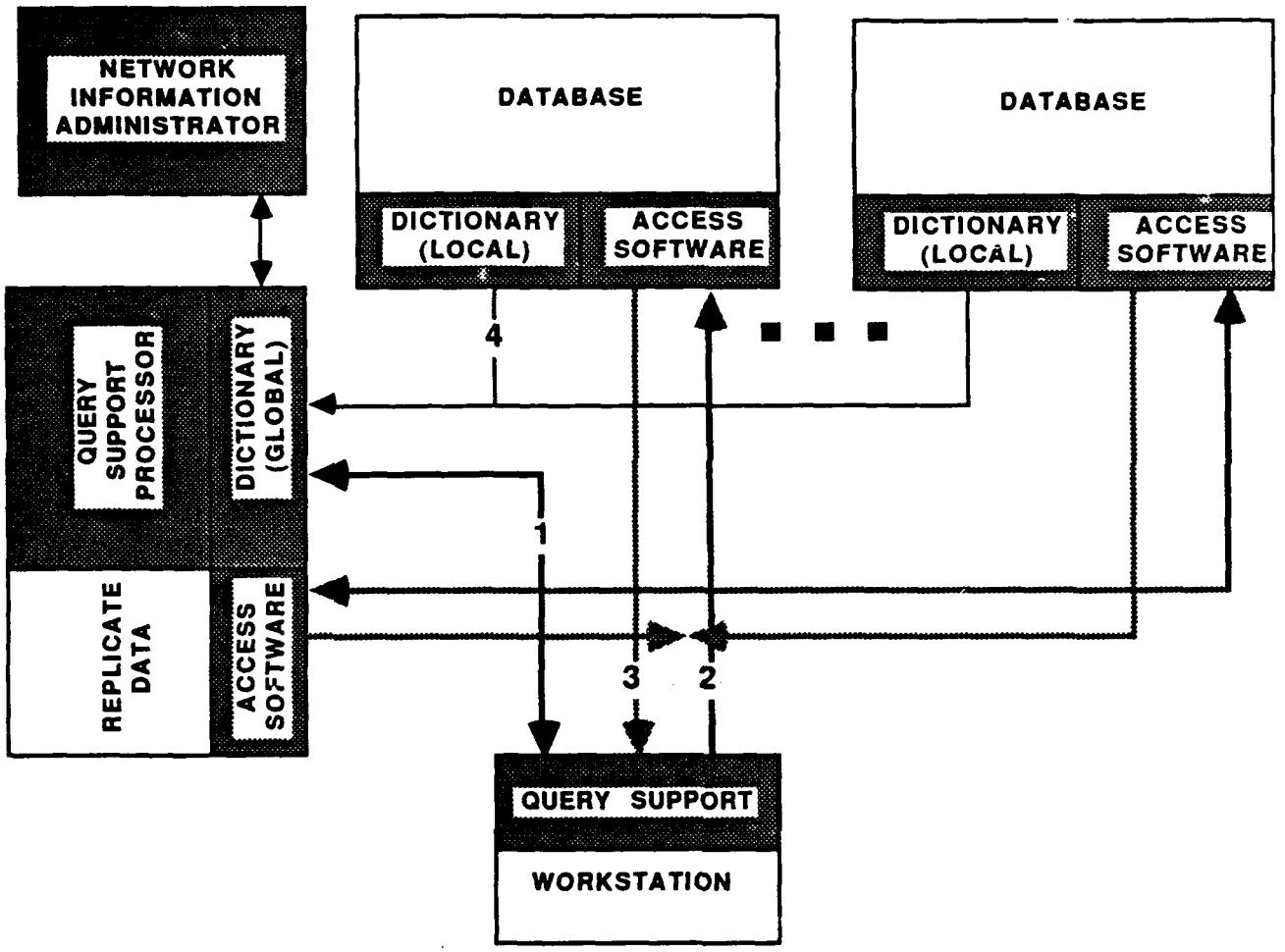

Figure 3

QSP Functional Relationships

status and execution of the query and subqueries are generated as products of this function.

Functional relationship 2 is the issuance of subqueries, which are independently routed to primary or alternate information sources. Subquery execution and status is monitored and managed by the QSP. Subqueries are received by the target databases, and host resident interface software utilizes the services of the target database system (to the maximum extent possible), to physically retrieve the requested data.

Functional relationship 3 is the transmission of subquery responses from the target databases back to the requestor. The QSP provides mechanisms to monitor the execution and status of the subquery responses and to conjoin a query response.

Functional relationship 4 is the interaction between the QSP, the Network Information Administrator, and the database administrators for the constituent database systems. Database administrators define subschemas,or user views, of their database for use by users of the network as a whole. Database subschemas for network users are exported to the QSP, where they are combined, by the Network Information Administrator, into a Network Schema. The network schema is analogous to a database schema. However, it is a representation of the totality of information available from the network. Logical subsets of the network schema, analogous to database subschemas, or user views of a database are defined for distinct classes of network level users. Since these subsets of the network schema encompass the totality of information available to specific groups of users at the network level, they are referred to as network views or network subschemas. The Network Information Administrator is responsible for the definition, modification, maintenance, and administration of the network schema and network subschemas. He is also responsible for the assignment of specific network subschemas to specific groups of users.

Linkages among data elements, data structures, applications, requirements, and their definitions, maintained by the QSP, can be utilized to precisely bound the scope of proposed application or data definition changes. The effects of proposed changes can be quantitatively assessed for multiple systems and user communities. System documentation can be interactively updated using automated tools driven by the dictionary in conjunction with application development. A formalized environment for development, that enforces standards and automates bookkeeping can be provided. Effective life cycle change and configuration management, naming convention and software standard enforcement should result in IDHS systems that are significantly more responsive to operational users, more flexible, and less expensive to operate and maintain.

Future Dictionary Directions

Data dictionaries began as manual tools for system analysis, oriented towards the initial implementation of computerized database systems. They evolved from passive reference tools to active, in-line development environments. Rome Laboratory is applying this technology at the network level to provide a network level development environment and to support network level access to information resources regardless of system location, database architecture, data representation model, or data format.

The future of this technology is in the sophistication and autonomy of the tools it can provide and the conceptual level at which they operate. Data dictionary technology today supports levels of information description that range from the physical connection media to the application program (i.e. data communication protocols) and from the information resource to the Information Resource Dictionary Schema Description. The remaining area to be addressed is at the conceptual level. In other words, how to extend the dictionary or its successor to accommodate abstract concepts and to reason about or act upon them. Examples include temporal, spatial, and event/subevent reasoning; and autonomous system self-optimization based on an internalized understanding of the characteristics of the data it processes, the needs of its users, where the system fits in the organization it supports, and where the supported organization fits into the external world.

Conclusion

The communication infrastructure required to support heterogeneous database access and distributed/decentralized processing is in place. Benefits of the Information Resource Dictionary to the database environment and at the network level are currently understood. The extension of Information Resource Dictionary capabilities to the network as a whole is technically possible and the proliferation of databases and increasing network connectivity make this essential. The result should be information systems that are significantly more flexible, responsive, and maintainable - with substantial cost savings and productivity increases.

BIBLIOGRAPHY

American National Standard for Information Systems, Information Resource Dictionary System, ANSI X3-138-1988, National Institute of Standards and Technology, July 1988

AOG Systems Corporation, Design Concepts for Database Utilities, RADC TR-86-48, April 1986,

Codd, E. F., The Relational Model for Database Management:, Version 2, Addison-Wesley, New York,1990

Goldfine, Alan, and Patricia Konig, A Technical Overview of the Information Resource Dictionary System, Center for Programming Science and Technology, National Institute of Standards and Technology, April 1985

Loomis, Mary E. S., The Database Book, Macmillan Publishing Company, New York,1987

Martin, James, Managing the Data-Base Environment, Prentice-Hall, Englewood Cliffs, NJ,1983

Ross, Ronald G., Data Dictionaries and Data Administration, AMACOM, New York,1981

Tremblay, J. P., and P. G. Sorenson, An Introduction of Data Structures with Applications, McGraw-Hill, New York,1976

Wiederhold, Gio, Database Design, McGraw-Hill, New York,1983

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.